

CSCE 236 Embedded Systems, Spring 2013

Lab 6

Thursday, April 4, 2013

Names of Group Members: *Note that the people you work with today will be those you will work with for the final project. Groups should be 2-3 people.*

1 Instructions

This is a group assignment to work on during class. You only need to hand in one copy of this, but make sure that the names of all of your group members are on this sheet to receive credit. Complete all of the sections below and make sure to get the instructor or TA to sign off where required.

2 I²C Inter-Robot Communication

In this section we will modify the sample I²C code from the Arduino library to enable bi-directional communication between two Arduinos connected together. The Arduino calls their I²C library *Wire*¹. Look at the Arduino example code for the Wire library, in particular the `master_writer` and `slave_receiver`.

Now, connect your Arduinos' I²C buses together. First, you must connect the grounds of the two boards together (to give them the same voltage reference). Then connect the SCL and SDA pins together. SCL is on pin A5 and SDA is pin A4 (they are also available above the AREF pin if you prefer). Test the `master_writer` and `slave_receiver` to verify that they function together correctly.

Checkoff: *Show the sample Arduino code working on your Arduinos.*

It turns out that with I²C a device can be both a master and a slave. Now, write code so that when one of the buttons is pressed, the light on the other Arduino will turn on. To do this, you simply need take the `slave_receiver` code and in the main loop add a transmission (same as from `master_writer`) whenever the button is pressed or released. You can use identical code on each Arduino, just make sure you switch the addresses. In addition to turning on the other's LED, print over the serial port "my button pressed" and "other button pressed."

Checkoff: *Show the code turning on and off the other Arduino's LED and the serial printing. Note that it should work from board A to B and from B to A.*

3 Range Finder

In this section you will start using a VCNL 4000 range sensor (it also has an ambient light sensor), which is an I²C device. The stated range for this sensor is 20cm, although it typically works best for measuring distances under 10-15cm. The datasheet and example code are linked to on the course website. But first, you must solder the sensor.

¹I²C is generically known as a "two-wire interface" (TWI) because using the I²C name used to require paying a licensing fee. SMBus is a stricter subset of the I²C protocol and is often used on computer motherboards to communicate with low-speed peripherals (e.g. a temperature sensor on the motherboard). In other words, there are three names (I²C, TWI, SMBus) that all refer to basically the same interface and protocol.

3.1 Soldering

Each person in your group must solder his/her own sensor as demonstrated in class. Please use the following colors so all sensors are consistent:

IR+ Red (hook up to 5V)
SDA Green
SCL Blue
GND Black
3.3 Yellow (hook up *only* to 3.3V)

Checkoff: *Each person must individually get checked off after soldering the board.*

3.2 I²C Communication Library

Download the code, datasheet, and application note for this sensor from the course website. Briefly read through these documents before trying the code. With the power disconnected, hook up your sensor to the Arduino. You will first need to determine which pins on the Arduino are the SDA and SCL pins. **Make sure you connect the yellow 3.3V line ONLY to the 3.3V supply on the Arduino and NOT to the 5V line.**

Test the sample code and perform some basic characterization of the sensor (for the final project you will need to do more).

Checkoff: *What value do you obtain when the sensor is very close (1cm) to an object? What about when it is further? Is the relationship between value and distance linear? What happens if you aim two sensors near each other (as could happen in the final project competition)?*

4 Final Project Preview

For the final project your team will compete against other teams in a modified capture the flag-type environment. Each robot will start guarding an exit of a “room” and the goal will be to get out of the opposite exit before your opponent². There will be a black line on the ground connecting the two exits that you can follow to the exit. There will also be walls at least 3 inches high surrounding the environment. You are free to modify your robots, but please make sure all of the components I gave you are returned in working condition. You must also fall within the following constraints:

- You robot must be fully autonomous (no user input once a game has started).
- You can use servos/sensors/etc from up to 2 of the robot kits.
- You can use other passive mechanical devices, but no other electronics/sensors/etc.
- The dimensions of your robot cannot exceed 20x14x14 centimeters and cannot be more than 500g.
- You can power your robot from either the wall power I supplied or batteries; however, be aware if you draw too much power from either of these your robot may reset itself.

Note that these are preliminary requirements and there may be minor modifications in the final project description.

There will also be a number of checkpoints associated with the competition. The first will be wall following using the range finder to follow a wall for at least 50cm. This checkpoint will be due in class on **April 16th**. Get started on this once you complete the above lab.

²If neither robot exits within the time limit, then the robot closest to its exit wins.